Theory of Computation

Prof. Kamala Krithivasan

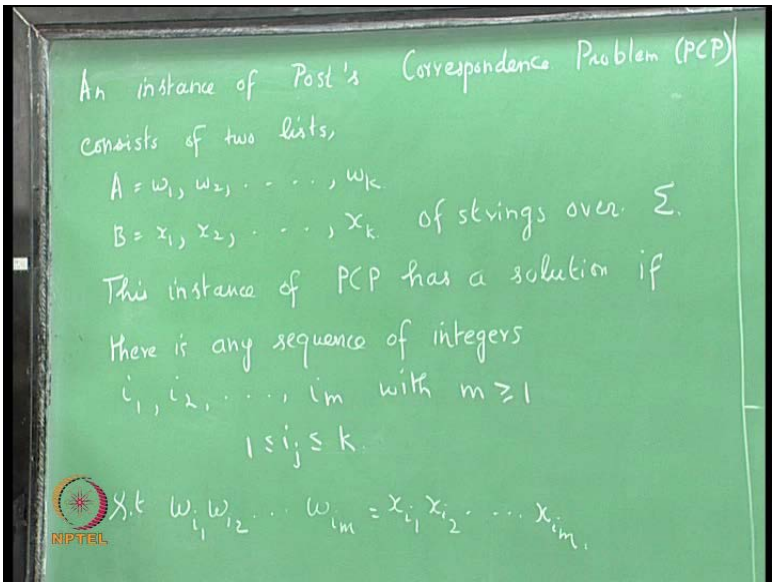
Department of Computer Science and Engineering

Indian Institute of Technology, Madras

Lecture No. #35

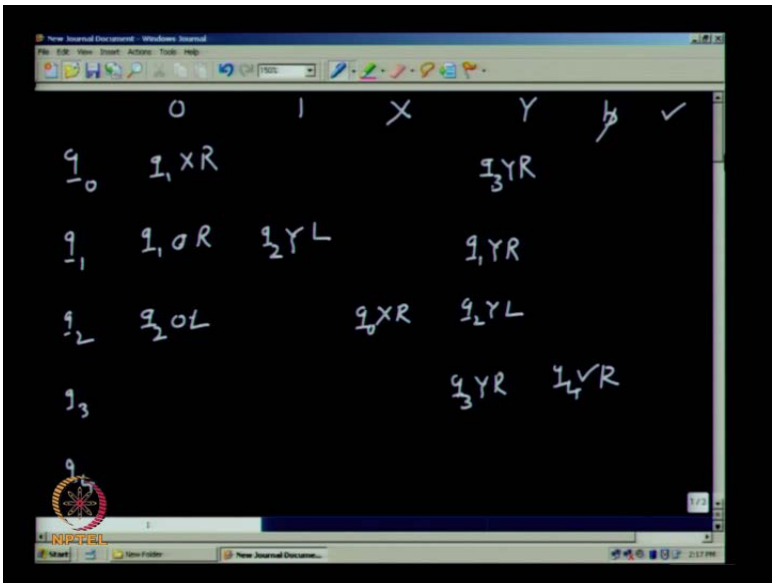
Post's Correspondence Problem(Contd), Time and Tape Complexity of TM

(Refer Slide Time: 00:15)



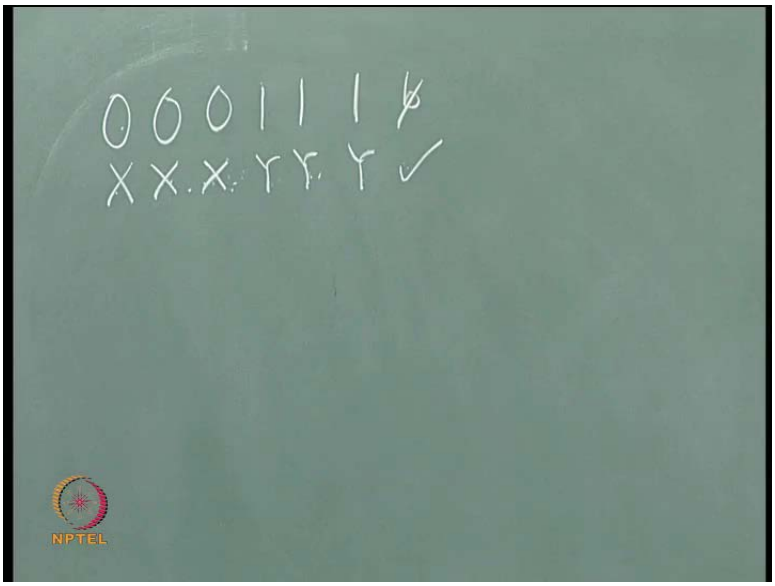
So, we were considering post correspondence problem.So,given two sets of words,the instance has a solution if you can find integer i 1, i 2, i n; such that w i 1, w i 2,w i m is equal to x i 1, x i 2,x i m.And given a Turing machine,how can you construct given m and w, how can you construct an instance ofM PCP,modified PCP?That is what we haveseen.

(Refer Slide Time: 00:41)



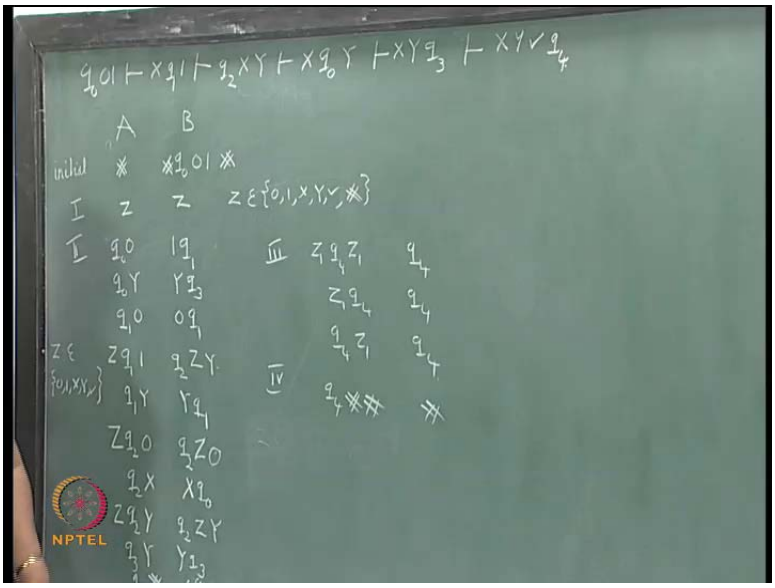
Let us take this example,this is a Turing machine which we have already seen which accepts 0 power n,it accept0 power n 1 power n.

(Refer Slide Time: 00:44)



And given a stringlike this,this changes into a Xagain goes here,changes this into a Ycomes back,changes this into a X,changes this into a Y comes back, changes this into a Xgoes,and changes this into a Ycomes back.And when it finds that there are no more 0(s), it has tosee that there are no more 1(s),and when in it encounters here, it puts a tick markand halts,saying that the string is accepted.

(Refer Slide Time: 01:27)



So, it has to accept01, this machine has to accept 01.What are the I D's?Q naught,the ID'sI will writelike the,q naught01,it changes that into X and goes toq 1.And when it sees a 1 in state q 1, it changes that into Y and moves to q 2, q2XY.And in q 2, when it seesX it moves right in q naught.And in q naught when it sees a Y it moves right in state q 3.In q 3 when it sees a blank it goes to q 4 and puts a tick mark and halt,q 4XY and then puts a tick mark and goes to q 4and then it has to halt.This is the sequence of IDs which leads you to acceptance.

Now, from the moves how do you construct two setsof words for the PCP?Two sets A and B,the first one is hashand q naught w,here q naught 01 hashthis is ainitial pair.Then group one will have every symbolif you have a 0, you have a 0; you have a 1, you have a 1; you haveaX, you have a X and so on.So,I will rather put it asZ Z, where Z can belong to01X Ytick mark.Hash,I left out,hash.Thengroup two should simulate themoves of the machine,group two pair of words.What are the pairs?q Naught0 is q 1X R.So, q naught 0 is q 1 X right.

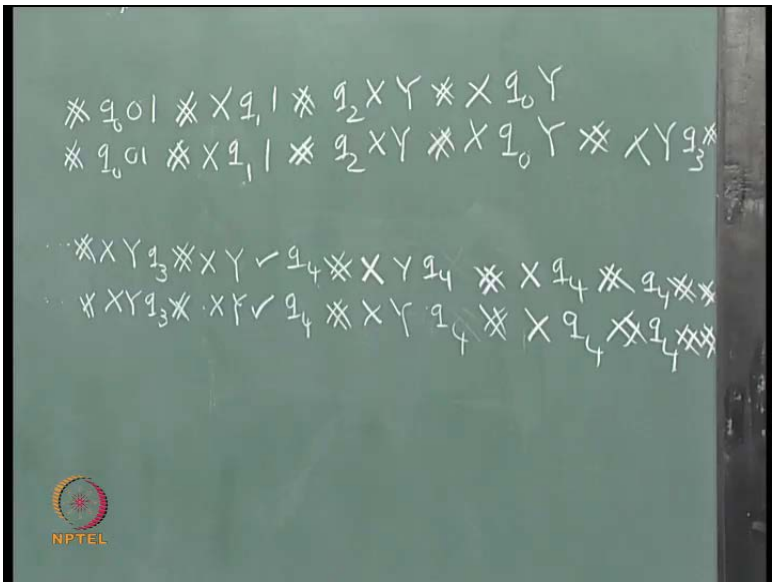
When the move is right there will be only one pair, when the move is left there will be several pairs.Then q naught Y is q 3Y Rq naughtY is y q 3R, there is only one pair because the move is right.If the move is left,you have severalpairs. q 10is q 10R the corresponding pair will be like this, then q 11 is q 2Y L, where again is it can beany one of this.Then q 1 is, q 1 Y is q 1 Y R,the pair will be like this.Then q 2 0 is q 2 0 L,q 2 0 is

q 20 L Z can be any one of the following symbols, butnot hash.Z can be here,Z belongs to 0 1X Y tick mark,for this one hash also I am including.Then q 2 0 is q 2 0 L then q 2 X isq naught XRand q 2 Y is q 2 Y R, q 2 Y is q 2Y L again Z can be any one of the symbols.

So, there will be several pairs corresponding to that.Then q 3 Y is q 3 Y R, q 3 Y is Y q 3and q 3 blank is q 4, q 3 blankfor that you have hash is q 4 tick markso, tick mark q 4.So, these are the pairs corresponding toZ can be,Z belongs to 0 1X Ytick mark it can be any oneany other symbol.So, these are the pairs correspondingto the moves.Then which is the final state? q 4 is the final state.q Naught it will change 0 to X and q 1 it will change 1 to Y,move left in q 2.And then if it sees y, it will move right in q 3 and finally, when it sees a blank it will stop in q 4.

So, the other sets of pairsare is any symbol Z 1 q 4 Z 1 q 4,Z 1 q 4 q 4 q 4 Z 1q 4 the last pair will be q 4 hash hashhash.Now, making use of these pairs of words, find out whether this instance?This is an instance of aPCP,this instance of PCP should have a solution if and only if m accepts w.What is w here,m is that machine,What is w? 0 1and 0 1 is accepted this is sequence of ID'sfor accepting.So, how will you builda two strings from these sets of pairs of words?So, that they are equal.

(Refer Slide Time: 09:28)



So, let us see how we can start?Start withhash and hash. The lower string will be longer.Each one will be one ID and then when you try to match the first string with the

second stringthe next ID will be created that is what we have seen.So, hash q naught 0 1 hash this is the initial pair, with this you startthen you have to create q naught 0 1(( )).So, q naught 0 when you want to create the first onethe corresponding pair is 1 q 1.So, q naught 0 when you try to create it will be 1 q 1 herebecause you have the pair q naught 0 1 q 1. Then any symbol is at Z you can add. So, 1 1 hash hash.So, when the first ID is created in the, initialID created in the first string, what is a next?I am sorry, 0 is changed into X so, this isX.

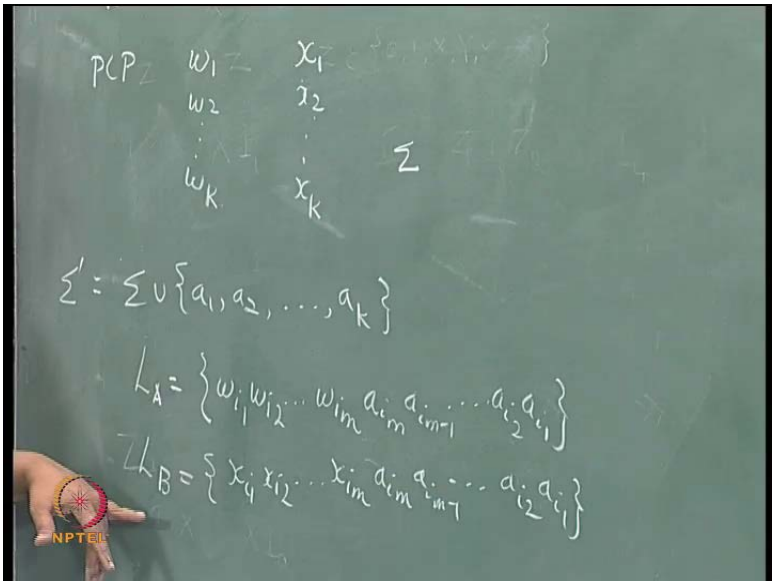
Sorry first move isq naught 0 is q 1 X R so, q naught 0 is X q 1.So, when you add q naught 0,you are adding X q 1 here.So, what is this ID X q 1 1? That ID is appearing there.Now, this ID has to be created hereso,X q 1 1 you have to create.So, what is the pair here?Z can be any one symbol we have seen,Z can be 0 1X Y tick mark.So, when you add X q 1 1 here,what you will add? q 2X Y.You will add q 2X Y then hash here and a hash here so, the next ID q 2X Y is created.Now, youmust try to create q 2X Y here.So, what is a move? q 2X is X q naught.

So, when you write q 2X?You have X q naught here.Then Y can written here Y can be written here hash can be written here a hash can bewritten here.So,X q naught Y is the next ID and that ID is created.Now,X q naught Y you have to create in the first string.So, you have to choose a corresponding pair q naught Y. q naught Y is Y q 3.So, first you add X, X X.You add q naught Y here, corresponding pair isY q 3 so,X Yyou, q 3 you got here,X q naught Y,the next ID X Y q 3 is created there.So, these two strings have come up to this (( ))that is the first string has come up to this.The second string has come up toX Y q 3 hash,I am continuing from there.

Now, what is the next move?q 3 hash is tick q 4.So, first you create X,you create aX here, you create a Y here,you create a Y here.Then when you try to create q 3 hash?You will createa tick mark and a q 4. q 3 hashit in this place, you would have got X Y tick q 4 hash.Now,we have reached the finall ID,you have to start consuming the symbols.So,X X Y Ytick q 4,just q 4 hashhash.Now, again it will start consuming the symbolsX X Y q 4,just q 4 here,one more symbol is consumedhashhash.Here,X q 4,but only q 4 here,then a hashhash.What is the last pair? q 4 hash hash so, you will have q 4 hash hashand only hashboth the strings have become equal.

So, you can see that successive ID's are created.And finally, if the string is accepted, the symbols are consumed and you end up with equal strings.So, the PCP has a solution,if and only if m accepts w.So, if PCP were decidable, if you have an algorithm for PCP?Then you can use that algorithm to find out whether m accepts w or notand whether m accepts w becomes decidable,but you know that that is not a decidableproblemso PCP is undecidable.

(Refer Slide Time: 16:44)



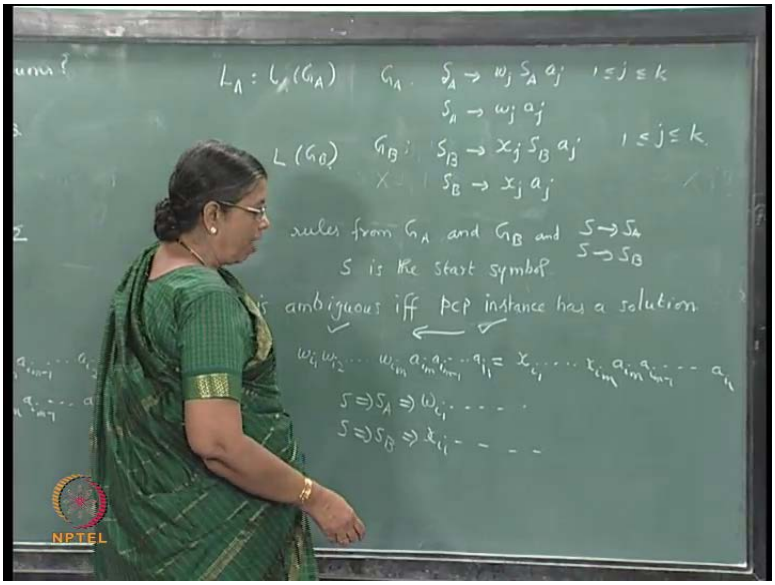
(No audio from 16:00 to 16:20)

Now, we can make use of the PCP to show that ambiguity problem is not decidable. (No audio from 16:27 to 16:33) The ambiguity problem is, isa context free grammarambiguous.Ambiguity problemisa CFG ambiguous. (No audio from 16:54 to 16:59)ACFG would be in this is input,input is CFG.Now, we can reduce PCP to ambiguity problem, how can you do this? (No audio from 17:18 to 17:25)An instance of PCP has two setsw 1 w 2w k two sets of stringsx 1 x 2x k.Now, you construct a,this is overan alphabet sigmaall this strings are over analphabet sigma.Now, consider an alphabet sigma uniona 1 a 2a k newsymbols, a 1 a 2 a k arenew symbols, there are not in sigma, some new symbols you have added.

Now, this alphabet you are considering,sigma dash is equal to sigma uniona 1, a 2, a k,corresponding to this k, you are having k's symbols a 1, a 2, a k.Now, consider a language L A, L A is a set of words of the form w i 1, w i 2,w i m,a i m,a i m minus

1upto a i 2a i 1strings of this form.Where i 1, i 2, i m they are all integers between one and k.And L B is a languagex i 1, x i 2,x i m,a i m,a i m minus 1,a i 2,a i 1. Now, this portion is from sigma this portion is made up of the new symbols.Each string has two portions,the first portion is made up of from symbols from sigma,second portion is made up of the new symbols.Now, you can have a linear grammar for L A, you can have a linear grammar for L B.

(Refer Slide Time: 19:45)



So, what is L A is actually?L Aof a grammar G A,language of a grammar G A.What is G A?G A has rules of the form.S A goes tow jS Aa j, j will vary from 1 to k,S A goes tow ja j. So, linear grammarand in the linear grammar you have k rules of the form w j, w j is generated on the left side, a j is generated on the right side.If and you can end up the derivation with it some w j.So, for each j between 1 and k you have a rule.So, you have two k rules like this. And using these two k rules, we can generate strings of this form see first you generate w i 1and a i 1then you generatew i 2,a i 2in between the non-terminalS Awill be there. Then you generate w i3,a i3 and so onfinally, S A will generate w i m a i m. So, this linear grammar will generate L A

Similarly, L B will be generatedbyanother grammar,S B goes tox j S B a ja andj will be between 1 and kand S B goes to xj a j.There will be two k rules for G Band with these two,it is a linear grammar with these two k rules.You can generate strings of the form x i1 x i2 x i m ai m a i m minus so on and so on.First x i1 and a i1 will be generated,then x

i 2and a i2 will be generated and so on.Now, consider a grammar Gwith the rulesfrom G Aand G B and two more rulesand rules of the form,S goes to S A, S goes to S B,where S is the start symbol.

(No audio from 22:34 to 22:45) Now, you can see that this G is ambiguous,if and only if, PCP instancehas asolution.Why in twodirections, you have to prove?Suppose PCP instance has a solution,then you have to prove that G is ambiguous and if G is ambiguous the PCP instance has a solution.Suppose PCP instance has a solution,then you will have w i 1 w i 2w i m this will be equal tox i1x i m,isnotit. There will be some sequence of integerssuch that w i 1 w i2w i m is equal to x i 1 x i 2 x i m.Then concatenate with a i m,a i m minus 1up to a 1.Here also use same stringa i m,a i m minus 1upto a i 1these,this is equal to thisand this is a same string.

So, they are equal.And this has a derivation from S A and this has a derivation from S B,is not it.This string will have a derivation from S A,this string will have a derivation from S B.So,S derives S A,derivesw i1etcetera.S derives S B,derives x i 1etcetera.So, there are two derivationsfor the same string and the derivations are leftmost it is linear grammar so, there is no leftmost, rightmost; everything is the same.So, you are to have two leftmost derivations for the samestring.So, that means G assuming that PCP instance has a solution,you are,you have proved thatG is ambiguous.

(( )) the problem isany givenG.

Any

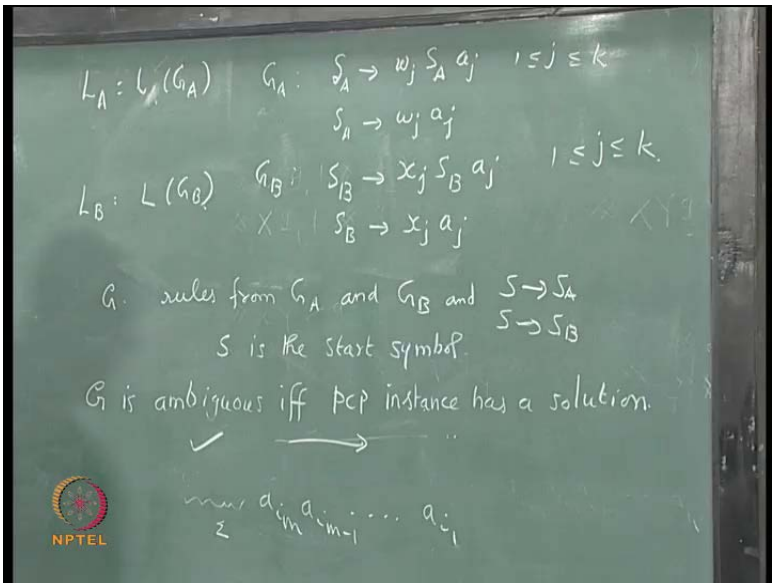
Any given grammar G is ambiguous.

No no we are constructing a grammar here

(( ))

See from the instance of PCP,I am constructing a grammar like that.Given this instanceof PCP,I am constructing a grammarlike that,And after constructing the grammarshowing thatG is ambiguous if and only if this instance of PCP has a solution.So, one way we have proved, that is if the PCP has a solution G is ambiguous.

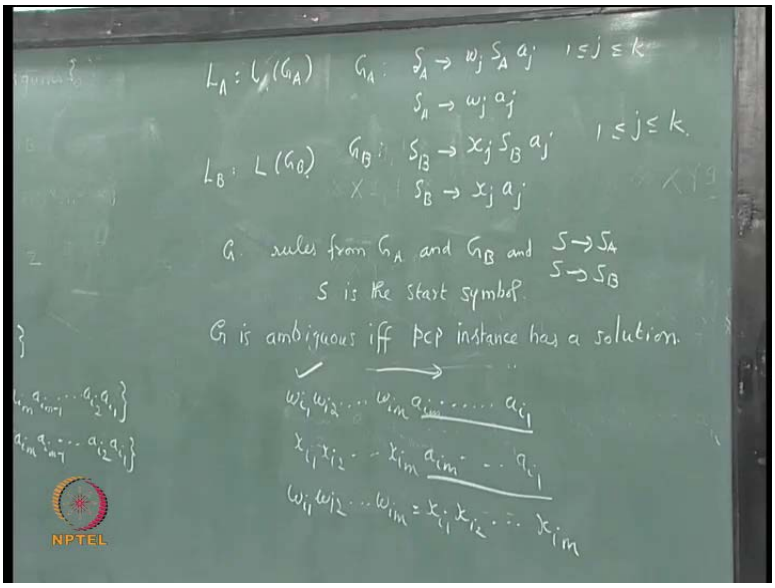
(Refer Slide Time: 26:19)



The other way aroundif G is ambiguousthe PCP has a solution.This what we have to prove?If G is ambiguousthen you have to show that PCP,this instance of PCP has a solution.Now, the string any string in the language hasone portion made up of sigma, another portion made up of the new symbols,some ai m a i m minus 1and a i1.And this sequence of symbols, determines the order of in which these rules are to be applied they are linear grammars.So, leftmost derivation only,only one non-terminal will occur in any sentential from.And this sequence, dictates the order in which the rules have to beapplied.If this is the sequence, later portion of the sequence, that meansshould have applied this S A goes to w i1S Aa i 1 first.

Then you should have applied S Agoes to w i2S Aai2 next and so on.Finally, last rule you would have applied S A goes to w i m a i m.So, this sequence of symbols in the later portion of the string,dictates order in which you have to apply the rules.So, any string ofthis form,first portion made up of the symbols from sigmaand the second portion made up of symbols, new symbols which you addedthat can be derived in only one way in S G A.That is there is only derivation possible for any such string in G A. Similarly, only derivation is possible in G B.So, if the grammar is ambiguousthat means there should be two derivations possible, at least two derivations.And we know that for any string only one derivation is possible in GA,only one derivation is possible in G B.

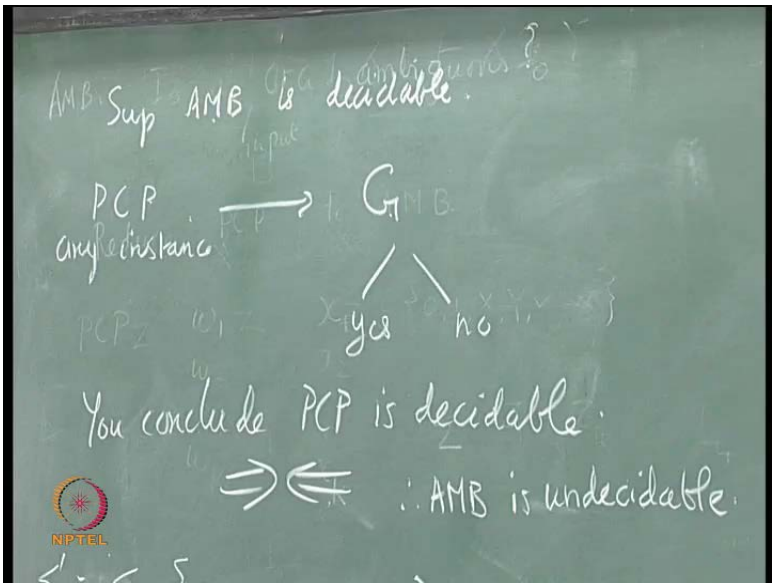
(Refer Slide Time: 28:55)



So, if there are two derivations that means there should be one derivation in G A,one derivation in G B.So,if you use that the given word G is ambiguous and thisw i 1 w i 2w i ma i ma i1is generated.Now, if this string is generated ambiguously there should be one derivation from G A,onederivation from G A. So, there should another derivation x i 1x i 2x i ma i ma i1in G B.They are the same stringsand this later portion you remove them, if you removeit is a same thing.If you remove, you find that w i 1w i 2w i mis equal to x i 1x i 2x im.That is you can find, sequence of integers i 1 i 2 i msuch that this equal to that.That is the instance of PCP has a solution.

So, givenone instance of PCP,you are able to construct a grammar Gsuch that G is ambiguous if and only if the instance of PCP has a solution.So, suppose the ambiguity problem, for G;ambiguity problem for context free grammar is decidable.Suppose we will write like that.

(Refer Slide Time: 30:35)



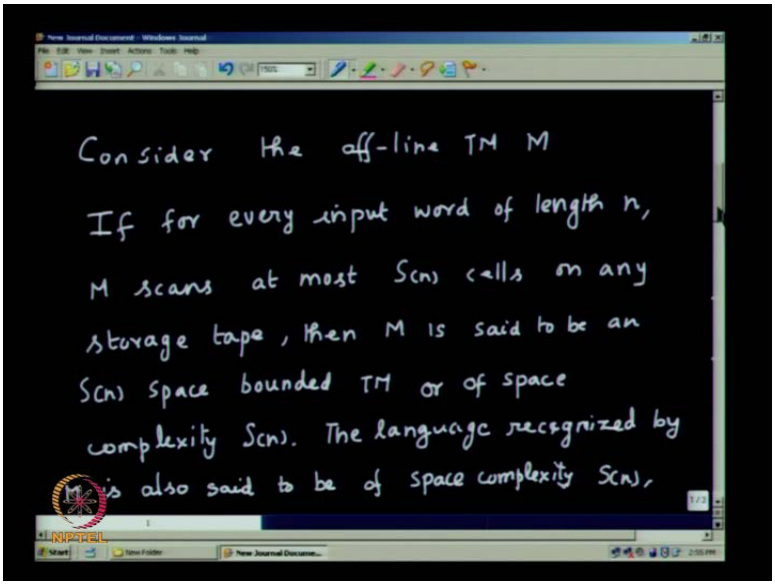
(No audio from 30:27 to 30:34) Supposeambiguity is decidable,then what happens?Any instance of PCP,any instancefrom that instance in this method constructG.Then ask is G ambiguous?If it is decidableit should say yes or no.Ask is G ambiguousthen you will come out is answer yes or no.If it is yes,PCP has a solution.If it is no,PCP does not have a solution, that means PCP is decidable.You concludePCP isdecidable,but that it is not correct,PCP is not decidable, you have already proved that.So, it is a contradiction.And the contradiction is becauseof the assumption ambiguity is decidabletherefore, ambiguity problemisundecidable. (No audio from 31:54 to 32:00)Many problems on grammars, you canshow undecidable like this.

For example, given two context free grammarsG1 and G2 is L G 1 equal to L G2is L G2containedin L G1.And given two grammars G 1 and G 2and regular set r some regular set ris L G1 equal to r is r containedin L G1 all such things are undecidable problems.And if many of them you will prove using PCP.It is easy, see actually the first problem to be shown undecidable is the halting problem.Then making use of that you have shown other problems undecidable.Where as far as grammars and strings are concerned,it is better to use some known problem,undecidable problem which is on strings,it is easier.So, first you prove PCP is undecidable,then reducing PCP to that new problem will be easy because PCP is on strings. So that way the proofs are given.

So, far about undecidability and decidability there are manyresult many more results,but nextwe will go on to complexity issues of Turing machines.And see what is meant?by an NP-complete problem?So, for that, first wesee what is a space and time complexity?See whenever you are writing a program, you are worried about the efficiency of the program.What do you mean by efficiency, how much time it will take?Usually time is taken as the measure of efficiency, butalso another problem which is also of importance is how muchmemory it will use.So, both issues memory, usage of memory is also important in some cases.Especially in some problems on computational geometry,memory also becomes a issue.

Because, sometimes you will be able to get good times measures by paying for that.In the sense that the space will increase a lot.There will be usually a trade off in some algorithmsnot saying all algorithms.In some algorithms when you try to improve on the time complexity.Number of steps, some because you have to use a proper data structure and so on.The data structure shouldhelp to search in a quick manner some and get some results.And keeping the data structure you may require large amount of space.So, when you want to improve on the time you will be paying in terms ofspace.So, both are important in certain cases, but mainly the time complexity is the one which rules over everything,I mean that is more important than the space.

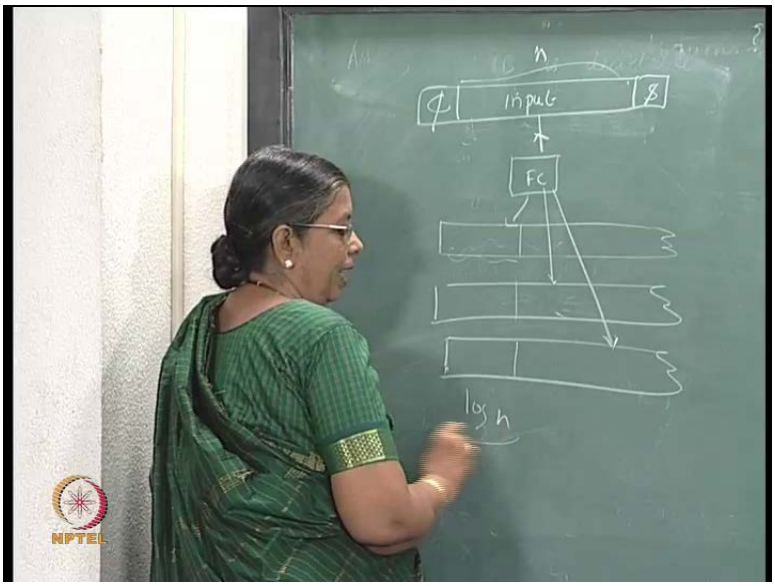
(Refer Slide Time: 35:12)



So, what do you mean by space complexity?We will start with terms of Turing machine.Another thing which we have been studying and insisting is that any Turing machine can be simulated by a random access machine or a register machine which is more like a computer. Turing machine is not random access because to get back to some information, you have to travel back and get the information.But, any Turing machine you can simulate with the random access machineand at most the time in (( ))t time is t squared and you can do that.And similarly, the other way round any random access machine you can simulate with the Turing machinebut time may at most be t square.

So polynomial or non-polynomial will not get affected because whichever modelyou follow.And this proof is not difficult,but I have not given the proof for the equivalence,it is given in the book.Sometimes later if there is time,I will do that otherwise.Simulation of register machine with Turing machine and Turing machine with register machine.It is that proof we can take as a reading assignment and read. Now, we will define time complexity in terms,space complexity in terms of Turing machines.So, what do mean by space complexity, what do you mean bytime complexity?

(Refer Slide Time: 36:49)

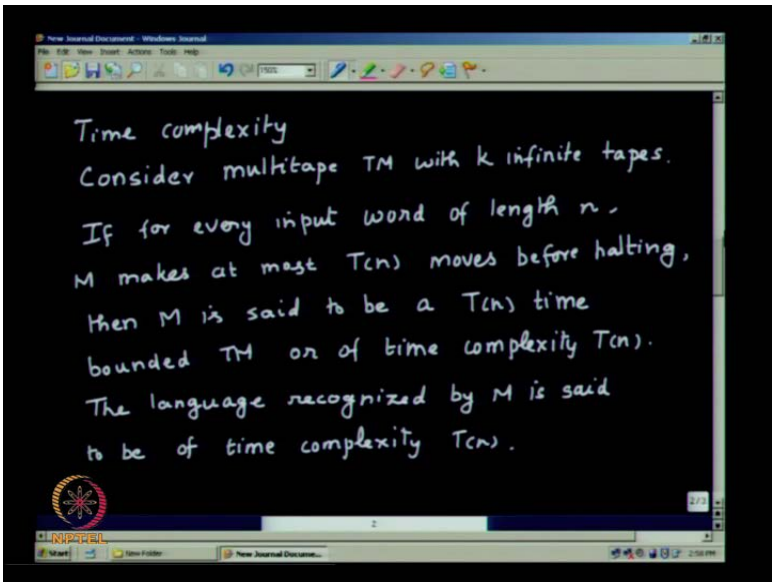


Now, the model you consider is slightly different for the two definitions.For the space complexity,you consider an offline Turing machine.What is an offline Turing machine? There is an input tapewhich is read only,there is a finite controland there are some tapeswhich can be used for reading and writing.Read write tapes, some tapes k tapes you

can have, each tape will have a tape head.The input is read only,you can moveon this,but within this spaceand on this, you can read and write on this tapes.The reason for separating the input is because the space if do not use a different tape,Input you have to anyway read.That means you have to go through all that space complexity cannot be less than (( )).

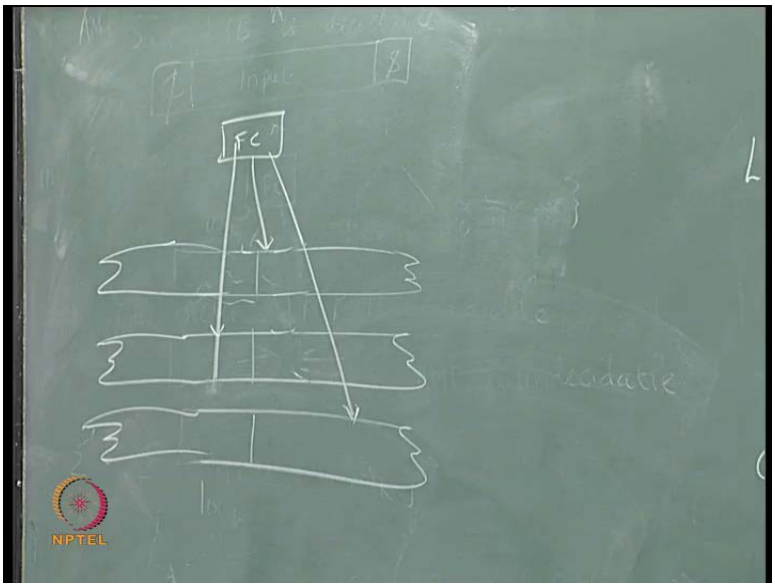
Whereas, herein for accepting this you may use less amount ofcells here,for example, log n's.If this is of length n,you may use just log n cellsin any one of the tape.And then accept it is possible to havespace complexity less than n that is why this definition offline Turing machine is given.Consider the offline Turing machine M,if for every input word of length n,M scans at mostS n cells on any storage tape then M is said to be an S n space bounded Turing machine or of space complexity S n.The language recognized by M is also said to be of space complexity S n.

(Refer Slide Time: 39:17)



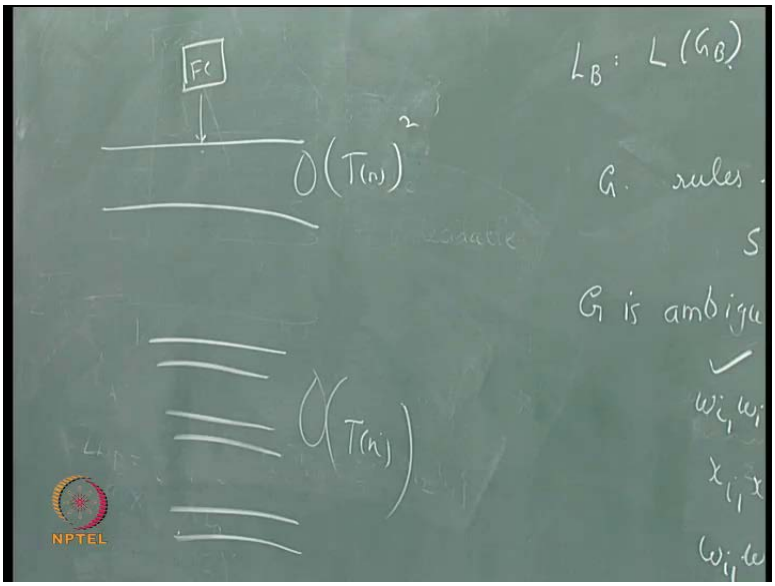
By separate the input, S n can be less than n also that it can be something like log n.If we are not separating the input, you have to read the whole input. So, space complexity will be n or more only, that is why this definition.Next, what do you mean by time complexity?In time complexity,you need not have to have separatething.Input can be,you can havea multi-tape Turing machinewith a finite control and tape heads.

(Refer Slide Time: 39:32)



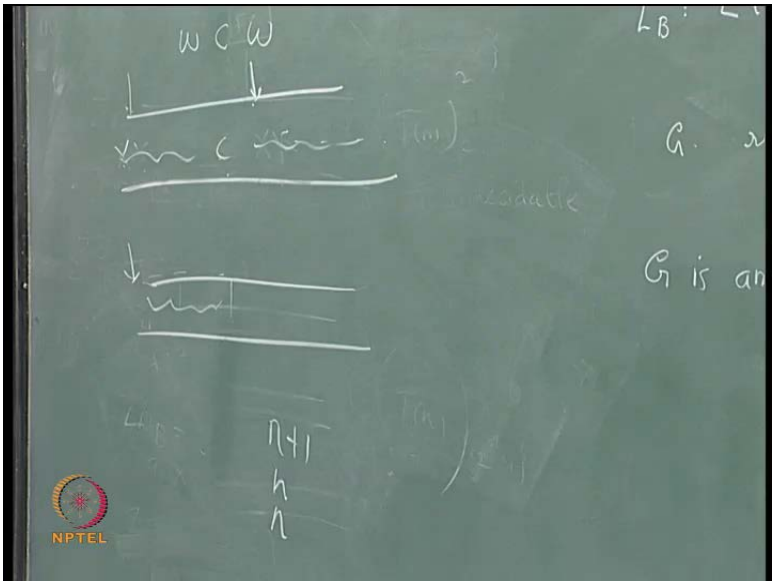
(No audio from 39:37 to 39:49) You need not separate the input out for a defining time complexity.Because, anywaytime complexity cannot be less than n,less thanall the cells you have to read one by one and then only accept.Consider a multi-tape Turing machine with k infinite tapes.If for every input word of length n,M makes at most T n moves before halting, then M is said to be a T n time bounded Turing machine or time complexity T n.The language recognized by M is said to be of time complexityT n.Now, it is important whether you are using multi-tape or single tape,you cannot just say single tape.

(Refer Slide Time: 40:48)



Because, we know thatif you have a single tape Turing machine, it can simulate a multi-tape Turing machine.Any single tape Turing machine can simulate,you have seen how the simulation can be done.By having,if you are having k tapes,you can have a single tape with two k tracks and so on.So, if the time is T n here,it will beT n squared here, of the order,general order.So, we just cannot say single tape Turing machine.You have to be careful whether the general definition is given for multi-tape Turing machine If you do it with the single tape,at the most you may be multiplying by another factor,if T n T n into T n factor.

(Refer Slide Time: 42:01)



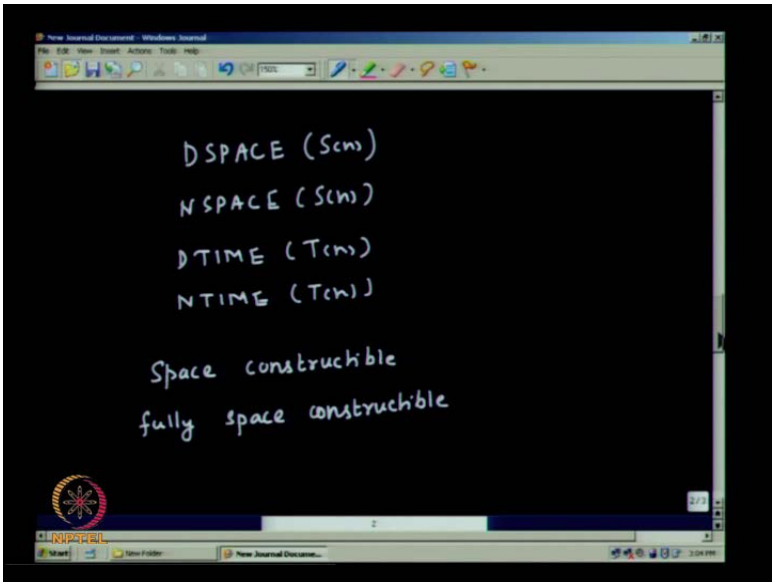
For example, you can seethat. (No audio from 41:53 to 42:00) I want to accept w c w.How will you accept w c w?In a single tapew is here, c is here, w is here.So, you will be marking the first symbol and checkingwhether it is the same symbol here.Come back mark the second symbol go andmark the second symbol after c and so on.So,you may have to make number of moves, if this isn 1 n length is n 2 n plus 1for checking one pair of symbols, it has to take n plus n,2 n moves and so on.So, the number of moves will be order n square.

Whereas, if you usetwo tape Turing machine.I have another tape also and w c w is given here.I want to check whether the first portion before c is the same as the second portion.In the second tape I will start from here and copy this w here.When I see the c, this pointergoes here,but this pointer will be moved here.So, going from here to here

andmaking a copy would have taken n plus 1 moves.Then moving it back to theside would have taken another n moves.Thenmove them simultaneously and check whether they are the same another n moves.

So, in 3n plus moves, the length is input, length is 2n plus 1in 3n plus 1 moves you will be able to check.If you have two tapes,if you have only one tape, you have to do it on order n square time.So, whether it is multi-tape or single tape is importantgeneral definition is given for multi-tape, Turing machine finite number of tapes.So, if for every input word of length n,M makes at most T n moves before halting, then M is said to be a T n bound Turing machine or of time complexity T n the language recognized by M is said to be of time complexity Tn.

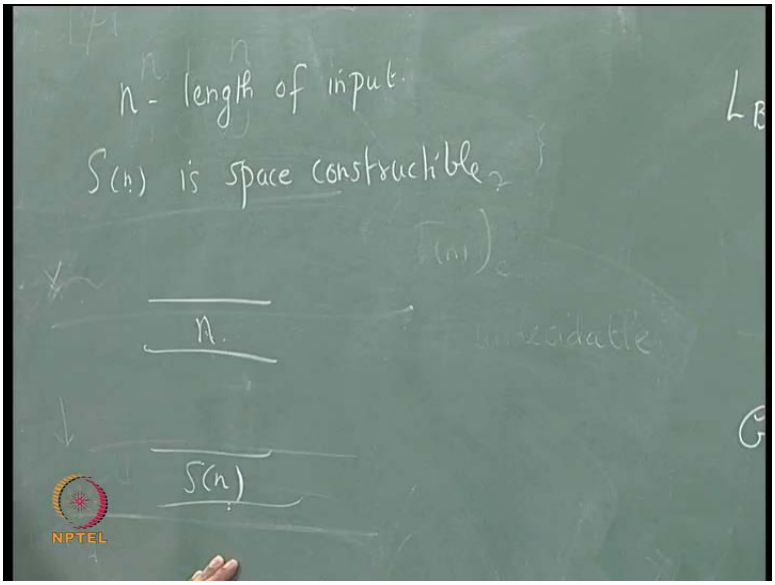
(Refer Slide Time: 44:43)



(No audio from 44:34 to 44:42) Then you talk about, see in these definitions conveniently we have said Turing machine without mentioning whether it is deterministic or nondeterministic.If you have a deterministic,Turing machineand it if it accepts a language with space bound S n that sort of class of language is denoted as DSPACE S n.DSPACE S n denotes the clause of languages accepted by deterministic Turing machine bounded S n space bounded Turing machine.Similarly, NSPACE S n denotes the clause of languages accepted by nondeterministic S n spacebounded machines.Similarly, DTIME T n denotes the clause of languages accepted by deterministic T n time bounded Turing machines.

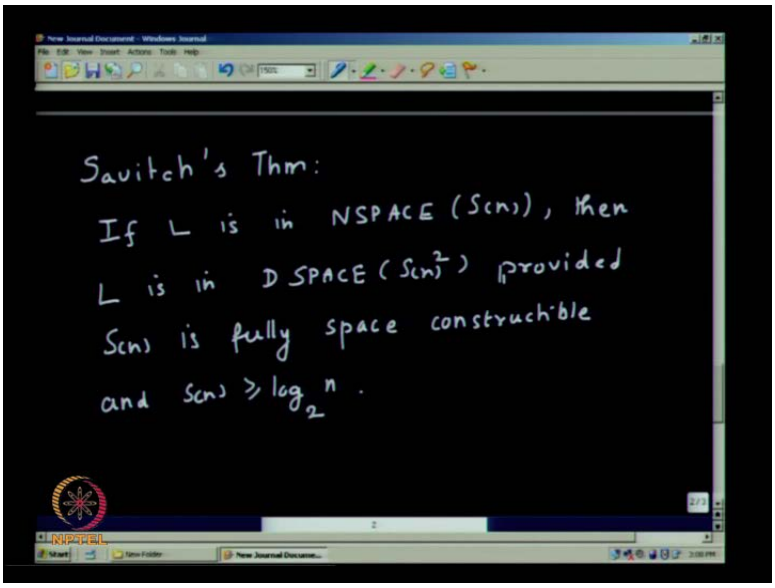
NTIME T n denotes the class of languages accepted by nondeterministic T n time bounded Turing machines. Now, we know that, what is a connection between S nNSPACE and DSPACE, DTIME and NTIME.Of course, that is themain thing,DTIME, NTIME you know that,when you want to simulate a deterministic Turing, nondeterministic Turing machine?With the deterministic Turing machine,the number of moves may increase exponentially.What about space?Again I will state some results withoutgiving the proofs.Proofs are in thebook,but main results I will just mention.Now, this T n or S n the function, that S n is a function of n isnot it,n is the length of the input.

(Refer Slide Time: 46:53)



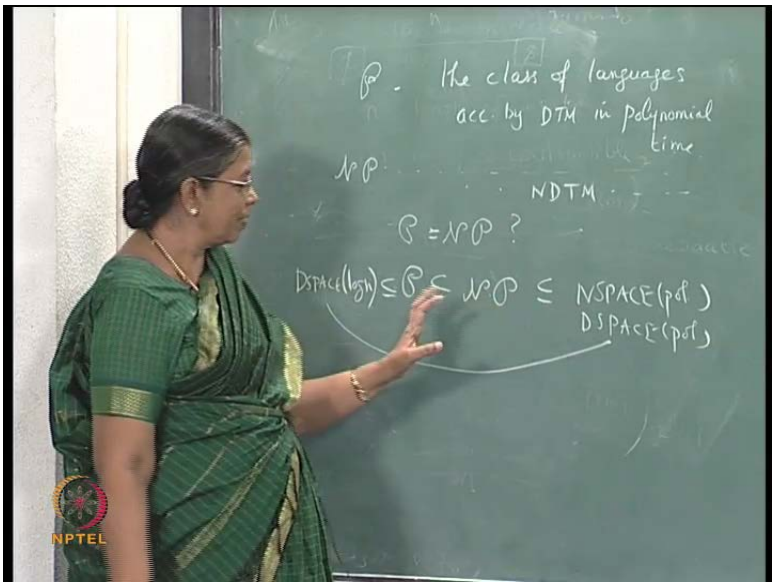
And s nis afunction is the space is denoted as a function of this.When do you say that function is space constructible,what is the definition of space constructible?S n isspace constructible. (No audio from 47:20 to 47:26)Suppose, for nthere may be several inputs of length n at least for one of them it should useS n space.Suppose, for all inputs of length n it uses much less than S n spacethen that S n is not space constructible.You follow, there should be one Turing machinewhichfor any n there should be at least one input of length n for whichthe machine useS n cells.Such a thing is called space constructible, such a function is called space constructible.

(Refer Slide Time: 48:42)



Now, there is an important result which is known as Savitch's theorem.This brings of the connection betweenif L is inNSPACE S n, then L is in DSPACE S n square provided S n is fully space constructible and greater than or equal tolog n to the base 2.You need not worry too much about this portionbecause many of the commonly known functions are space constructiblelog n,2 power n, any polynomial; n factorial to everything is space constructible.So, you need not worry about this portion,S n is fully,this you need not worry too much.The main point is this, if L is in NSPACE S nthen L is in DSPACE S n square.

(Refer Slide Time: 49:51)



(No audio from 49:39 to 49:47) Generally what is P P denotes the languages accepted by deterministic Turing machine in polynomial time.The clause of languagesaccepted by Ddeterministic Turing machineinpolynomial time.NP is a clause of languages accepted bynondeterministicTuring machine inpolynomial time.This still it is an open problem by theP is equal to NP is an open problem.Now, what can yousay aboutn spacepolynomial?Suppose, there is a polynomialPNit can be accepted by deterministic Turing machine in space.

So, polynomial, this is a polynomial, this also a polynomial.So,if it is polynomial,they are equal,NSPACE and DSPACE are equal.So, in the hierarchy any wayP is equal to NP or not is known,but definitelyP is containin NP,whether is proper inclusion or not that is the question.Now, this will be containedinn spacepolynomialis not it, this and DSPACE polynomial are equal. Now, DSPACE log nwill be contained in P.Because, in any tape, if you use only log n cells, how many number of times you use?It will be still polynomial time.So, this will be contained in this and by a space hierarchy results. Thisis properly included in this,but these inclusions whether they are proper or not, we do not know.

But at least one must be proper inclusion because this one is properly included in this.It looks as a this properinclusion but it has notbeen proved.So, with the brief introduction to this sort of an idea,we shall consider, what is meant by a complete problem?For a clause,for any clause what do you mean by a complete problem, and in general what is an NP-complete problem, and cook’s theorem in the next lecture.